# ND on wireless links and/or with sleeping nodes

Problems

draft-vyncke-6man-mcast-not-efficient
draft-yourtchenko-colitti-nd-reduce-multicast
draft-chakrabarti-nordmark-6man-efficient-nd

# Context

# Original assumptions in RFC 1970 [1996]

- Based on shared medium (10BASE5, 10BASE-T with hubs)
  - Multicast as reliable as unicast
  - The network cost of multicast is the same as unicast (*total receiver cost is higher unless filtered in NIC*)
- Nodes are always on
  - The effort to power on a host was not optimized
  - Simple to multicast DAD probes to find duplicates

# Sleeping nodes [RFC 6574 from IAB workshop]

- Small, low-cost, battery-powered nodes often "sleep" to extend battery life
  - Keep *just enough* of the system on to wake up on schedule
  - Radio **receiver** and transmitter are off
- Wake up periodically to perform functions
- Before transmitting to the network
  - Detect Network Attachment (DNA) by ucast NS to routers
  - Sometimes also DAD for LLA & global, then MLD reports for all solicited node mcast (hence at least 3 mcast packets if EUI-64, more if RFC 4941)
- Different than battery saving with wakeup on packet reception
  - Suggest using "battery nodes" term for those?

# Radio Efficient Nodes [IEEE 802.11 Low Power Wi-Fi clients]

- Even if main processor CPU is sleeping, try to keep radio going
- Radio is shut between AP beacons (100 msec)
- WiFi AP stores:
  - unicast frames destined to LP nodes
  - All multicast frames
- LP clients wake-up to listen to on AP beacons
  - Traffic Indicator Map (TIM) indicates whether to poll the AP to collect the frames
  - 1 bit states whether one or more multicast stored frames follow the beacon frame

# History

- Problems raised in IAB Smart Object workshop (Mar 2011)
- ND problems stated with initial solution at IETF 82
  - draft-chakrabarti-nordmark-energy-aware-nd
  - Updated and renamed based on WG feedback since then
  - Most recent update with more details and additions based on WG feedback to handle VRRP and router state loss
- Problem statement for WiFi multicast
  - draft-vyncke-6man-mcast-not-efficient this IETF
- Tuning and minor changes to RFC 4861
  - draft-yourtchenko-colitti-nd-reduce-multicast this IETF
- Email suggestions to reuse and extend DHCP
  - No draft

# Problems

# Problem areas

- Multicast [on WiFi, battery/sleepy nodes)
  - Wasting bandwidth
  - Waking up hosts unnecessarily
  - Looking at ND (RS, RA, DAD, address resolution)
- Duplicate address detection
  - Currently requires always-on to defend address
  - Requires waiting for 1 second for response (no “OK” response)
- Related DAD issues
  - Not robust against packet loss
  - Has loopback issues (see enhanced-dad draft)
  - Deployment issues for N:1 VLAN model in DSL (see dad-proxy draft)

# Wi-Fi Multicast Background

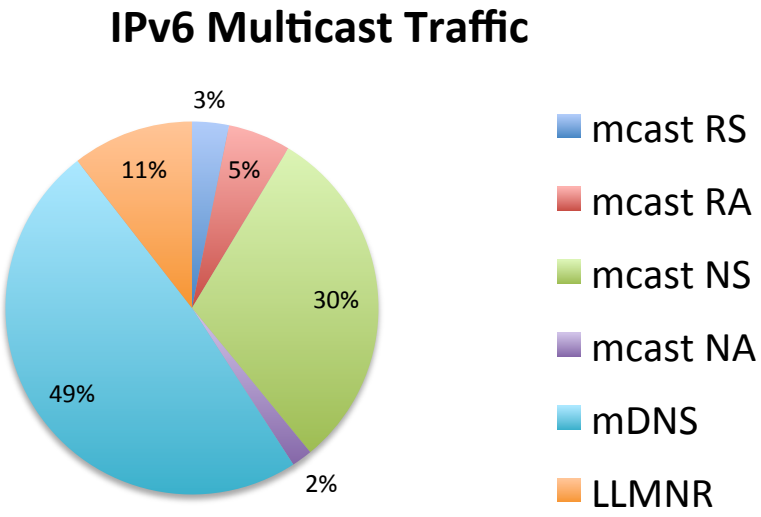
- Radio is a shared media: ucast, mcast and bcast frames require exclusive use of the media during transmission
- Additional 802.11 headers + management overhead
- Only unicast frames are acknowledged and retransmitted:
  - 10% packet loss appears to be common
  - Therefore, 10% of multicast frames are lost
- Depending on radio conditions, each Wi-Fi client has its own radio rate
  - => AP must transmit bcast/mcast frames at the lowest possible rate to ensure good reception
  - Makes bcast/mcast up to 10x more expensive than ucast:
    - IEEE 802.11a mcast: 6 Mbps, ucast up to 54 Mbps
    - IEEE 802.11n mcast: 15 Mbps, ucast up to 150 Mbps

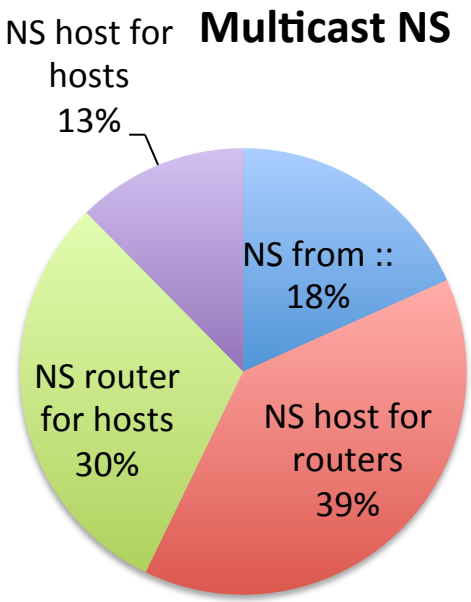
# RFC 4541: MLD Snooping is not Enough

- Solicited-node multicast **was a good** idea:
  - MLD snooping could filter in switches
  - NICs could filter multicasts
  - Works best with EUI-64 based IPv6 addresses (same group)
- RFC 4541 switches
  - Implement MLD snooping for global mcast
  - What if MLD report is lost?
  - Cannot economically/physically implement RFC 4541 for solicited nodes mcast esp after RFC 4941 (privacy addresses)
  - Results in flooding the mcast NS
- (v)NIC filters help with battery lifetime
  - Do not help with multicast bandwidth use
  - How many layer-2 mcast addresses can a (v)NIC support before interrupting the CPU?

# Some data from IETF-hotel Wi-Fi

- Collected by a mostly silent node in promiscuous mode, 75% of IPv6 traffic was multicast



# Factors to consider

- Looking for general applicability
- Looking for incremental deployability
  - work on links with existing hosts and routers

- Different scope and time frames:
  - Operational advise (setting timers etc)
  - Implementation advise
  - Small protocol changes – *what can we gain?*
  - Implicit or explicit registrations?

# Other problems in the neighborhood

- Need to be aware of relationship with e.g.,
  - SAVI [RFC 6620] nodes/routers/APs being deployed
  - ND DoS (scan all of /64 to use up memory for incomplete NCEs)
  - Hosts that frequently pick new addresses e.g., on each wakeup – no “leave/unregister” to remove from neighbor cache

# 802.11

- AP informs the cell in each beacon about Basic (mandatory), supported and disabled rates.
- Unicast:
  - Host tells the AP when it is sleeping/awake
  - AP buffers packet until host is awake
  - Frames retransmitted if not received (ACK system)
  - Highest rate the AP/host supports
- Multicast
  - Sleeping hosts synchronized to DTIM beacons
  - AP includes DTIM in beacon, which triggers all hosts on the link to stay awake for a period waiting for the multicast/broadcast frame
  - No retransmits / no delivery confirmation system
  - Transmitted at one of the common rates (Basic). E.g. unicast might be transmitted at 1.3Gbps while multicast is transmitted at say 24Mbps.
  - A consequence of sending multicast slow, is that it stays longer in the air, and is more susceptible for RF interference. E.g. a frame spends 3 ms in the air transmitted at 12Mbps, while only 120us sent at 300MBps.
- Host sourced multicast:
  - Sent L2 unicast to AP
  - AP multicasts the frame back out on the link

# Problematic ND behavior

- Any multicast message requires all hosts on the link to stay awake for 1-10ms (that's a long time). Even when the message is not for them.
- Multicast transmitted at a low rate, eats up a lot of time on the link
- Multicast being unreliable (8-10% packet loss or more), means you cannot trust DAD to detect collisions. (a collision requires two multicast packets to make it through. With 10% packet loss there is a 20% chance of not detecting a collision)
- Uses multicast for host to host communication